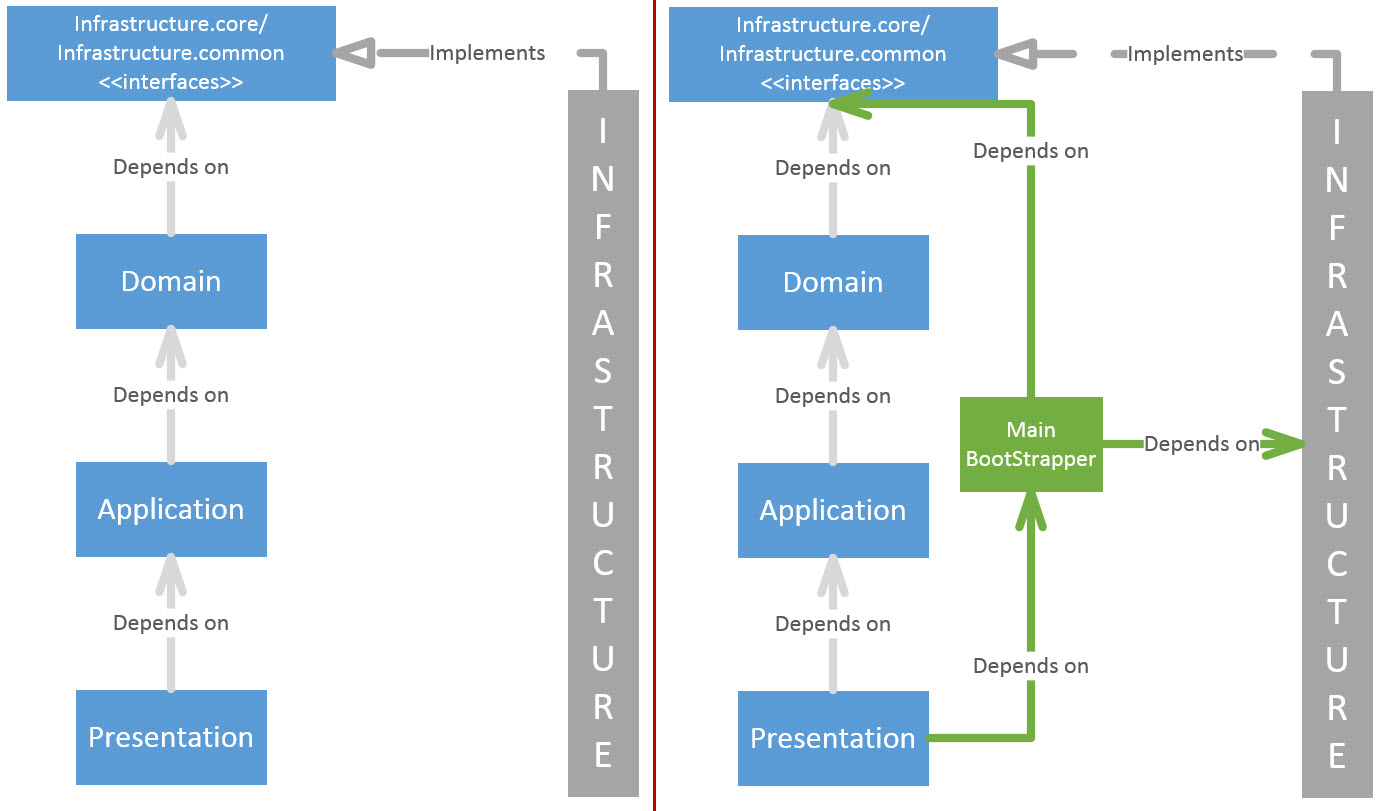
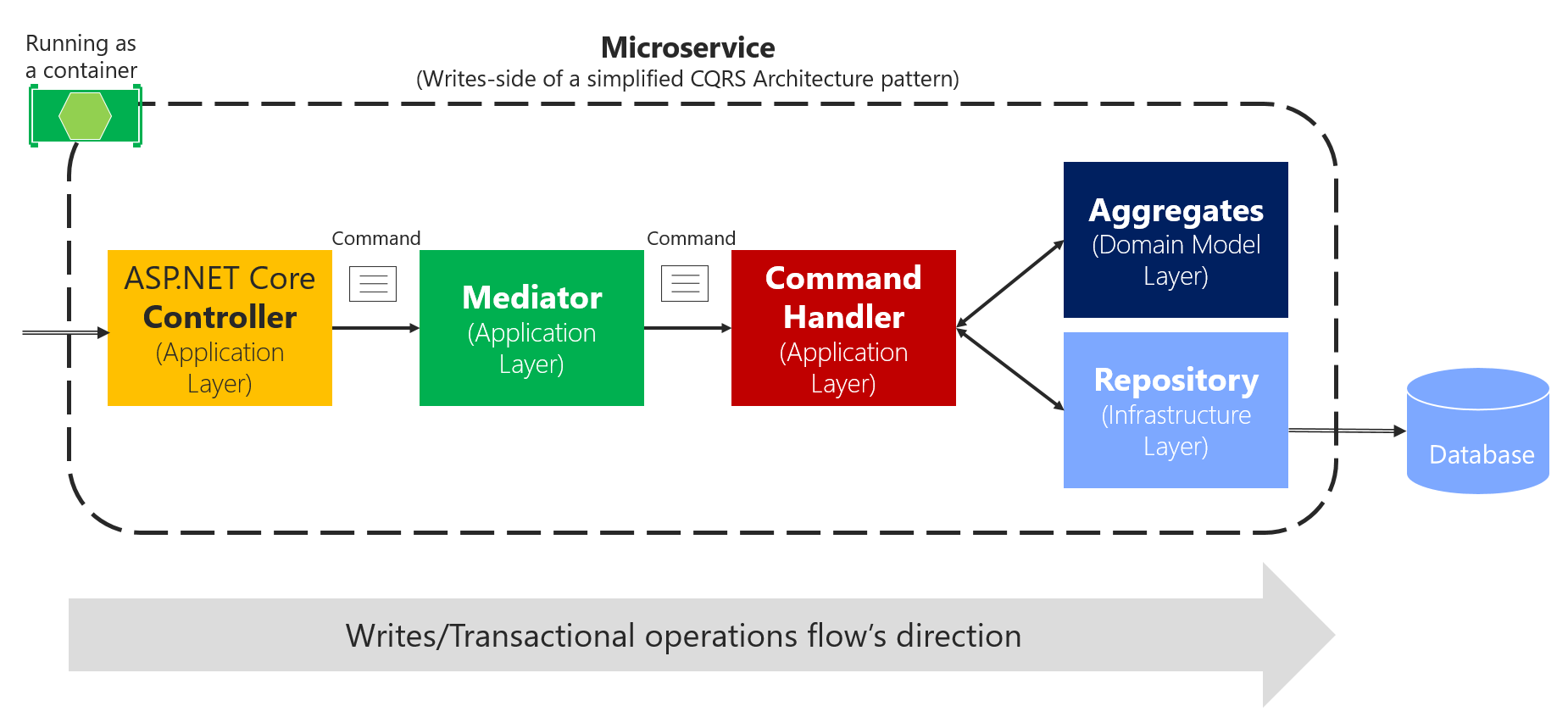
1/ Clean Architecture

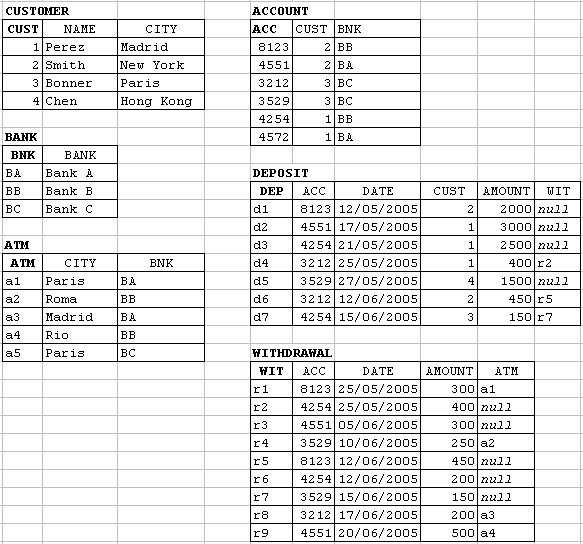


2/ CQRS and MediatR



3/ Database:

1. From “ChiTrung.Infra.Data” project, please change the connection string in “appsettings.json” file to your connection’s information. Ex: "DefaultConnection": "Server=yourmachine\\sqlexpress;Database=ChiTrung;Trusted\_Connection=True;MultipleActiveResultSets=true"
2. Database Migration (Code First): There are 2 migrations “Initial” and “AddBanksTable” but you should run the command “Update-Database AddBanksTable -context ChiTrungContext” to get completely database
3. Database model:



**Withdrawals of Funds:**

There are three types of withdrawals of funds, ATMs’ withdrawals of funds, transference of funds between accounts and purchases of products or services.

Funds transfer:

* In a funds transfer we have a withdrawal of funds from one account and a deposit of funds toward another account for which the amount should be the same.
* In a funds transfer, the withdrawal code from where the funds came is inserted in the Relation Variable DEPOSIT.
* The account’s owner of the withdrawal of funds should be the same as the customer who deposits the funds.
* In funds transfer there does not exist the ATM’s code.
* In this context I default the amount each account is unlimited for the easier demonstration

1. Run project: Applied default login/register/logout of aspnet core so please register and login before register new customer and add information for tables which belong to a Bank.
2. Applied rowversion (optimistic concurrency) and unique constraints for register new customer
3. Applied transaction locks (pessimistic concurrency) but aspnet core 2 doesn’t support transaction by native. Please take a look function CommitTransaction in the UnitOfWork.cs file. (I’m not using it now)
4. Please install Visual studio 2017 community preview 15.3.0 for using aspnet core 2.0